Rule-based Real-Time ADL Recognition in a Smart Home Environment

George Baryannis[1], Przemyslaw Woznowski[2], and Grigoris Antoniou[1]

[1] Department of Informatics, University of Huddersfield, Huddersfield, UK
{g.bargiannis,g.antoniou}@hud.ac.uk
[2] Faculty of Engineering, University of Bristol, Bristol, UK
p.r.woznowski@bristol.ac.uk

Abstract. This paper presents a rule-based approach for both offline and real-time recognition of Activities of Daily Living (ADL), leveraging events produced by a non-intrusive multi-modal sensor infrastructure deployed in a residential environment. Novel aspects of the approach include: the ability to recognise arbitrary scenarios of complex activities using bottom-up multi-level reasoning, starting from sensor events at the lowest level; an effective heuristics-based method for distinguishing between actual and ghost images in video data; and a highly accurate indoor localisation approach that fuses different sources of location information. The proposed approach is implemented as a rule-based system using Jess and is evaluated using data collected in a smart home environment. Experimental results show high levels of accuracy and performance, proving the effectiveness of the approach in real world setups.

Keywords: Event Driven Architectures, Activity Recognition, ADL, Indoor Localisation, Smart Home, Multi-Modal Sensing

1 Introduction

In the last two decades sensors have become cheaper, smaller and widely available, residing at the edge of the Internet. A single sensor provides only partial information on the actual physical condition measured, e.g. an acoustic sensor only records audio signals. A single measurement may be useful for simple applications, such as temperature monitoring in a smart home and may be sufficient to discover very simple events, such as fire detection. However, it is often insufficient for an automated *Activity Recognition (AR)* system to infer all simple and complex events taking place in the area of interest. Therefore, a fusion of multiple sensor-related, low-level events is necessary.

The Internet of Things (IoT) paradigm offers an effective way of acquiring and delivering low-level sensor events. The strength of IoT lies in the foundations of the Internet i.e. distribution of resources, support for common naming schemas and ontologies, common access strategies and availability of computational resources, to mention a few. The challenge is to locate and fuse the right pieces of (sensor) information in order to realise AR at the best quality of information possible. There are multiple ways of approaching sensor-based AR.

Chen and Khalil [3] propose a broad categorisation into data-driven approaches, exploiting machine learning techniques, and knowledge-driven approaches, leveraging logical modelling and reasoning. Both directions have their strengths and weaknesses. Machine learning techniques are criticised for not handling data conflicts well and for requiring large, annotated training datasets, while logic-based approaches are not as robust against noise and uncertainty and require carefully crafted rules.

In a multi-modal smart home environment AR usually focuses on the so-called *Activities of Daily Living (ADL)*, with the purpose of supporting *Ambient Assisted Living (AAL)* efforts, either for long-term monitoring of health-related features or for direct assistance. Such a setting brings about several requirements, such as the increased need for robustness against noise due to multiple sensors and the support for complex, uncertain and non-sequential scenarios [6]. Additionally, the user's location within the home must be recognisable with minimal user involvement (e.g. without requiring them to carry or wear a device). Inference of real-time, continuous streams of meaningful and actionable events is also a prerequisite for ADL assistance [4]. Finally, smart homes increase the difficulty in acquiring training data, since data are environment-dependent [10].

In this paper we propose a novel rule-based ADL recognition system, which is capable of reasoning over historical and real-time, multi-modal sensor data acquired in a smart home environment used as an experimental testbed. Reasoning is applied in a bottom-up, multi-level manner to support complex ADL scenarios, while rules employ non-deterministic patterns to account for missing activities. The system is capable of correcting erroneous sensor data through encoding of simple heuristics (based on expert knowledge) and cross-validating sensor readings against other sensing modalities. Such 'cleaned up' and fused sensor data are then used to achieve indoor localisation and ADL recognition. Experimental evaluation shows that high levels of accuracy and performance are achieved, in both offline and real-time modes.

The rest of this paper is organised as follows. Section 2 gives an overview of the smart home testbed that motivates our research. Section 3 provides an analysis of the offline ADL recognition system, while Section 4 details the modifications applied for the system to also work in real-time. Section 5 offers details about the system implementation as well as the results of the conducted experimental evaluation, Section 6 compares our approach to the most relevant ones in literature and Section 7 concludes and points out topics for future work.

2 Background

2.1 Experimental Testbed

Existing AAL systems make use of (environmental) sensor networks, wearable devices and computer vision technologies. Some research projects focus on a single sensing modality, while others, such as ENSAFE[3] and eWALL [8], implement

[3] `http://www.ensafe-aal.eu`

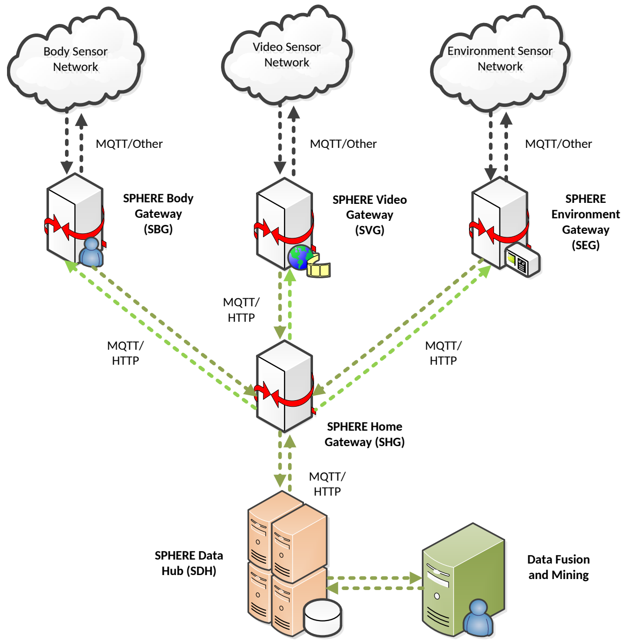

Fig. 1. An overview of the SPHERE system architecture [14]

multi-modal AAL environments. The SPHERE (Sensor Platform for HEalthcare in Residential Environment) architecture attempts to combine different sensing technologies to provide a generic platform for ADL recognition. This generic, multi-modal sensor-based platform, which has been built on cutting edge infrastructure made up of commercial and prototype components, will be used to test clinical and health related hypotheses in a real life environment. The sensor-based platform has been deployed in a two-storey, two-bedroom house, converted into a fully-instrumented living lab referred to as the *SPHERE house*.

The SPHERE platform is based on three sensing technologies: an Environment Sensor Network made up of hardware sensing the home ambience; a Video Sensor Network, relying on RGB-D cameras deployed in specific rooms in the SPHERE house; and a Body Sensor Network made up of ultra low-power, wrist-wearable sensors. Environmental sensors specifically include: temperature, humidity, passive infrared (PIR) and door contact sensors; light, noise and air quality sensors; and water and electricity meters. Fig. 1 provides a high-level view of the SPHERE hub and data sharing system. A detailed description of the system architecture and deployed sensors can be found in [14], along with a comparative analysis of similar multi-modal sensing platforms.

2.2 ADL Ontology

In order to have a common, controlled vocabulary for any ADL-related effort in SPHERE, (e.g. data generation, ADL recognition, annotation of ground truth videos), an ontology has been defined, listing and categorising activities occurring in the home environment. It was developed with the explicit aim of compliance with existing models, to achieve interoperability and applicability of collected datasets beyond the project. It is based on BoxLab's Activity Labels[4] and thus extends their model. A detailed presentation of the SPHERE ADL ontology can be found in [15]; the latest version is available in the OBO[5] format from http://data.bris.ac.uk (DOI: 10.5523/bris.1234ym4ulx3r11i2z5b13g93n7).

3 Offline ADL Recognition

The initial version of the proposed ADL recognition approach allows for offline analysis of activity patterns in a residential environment. Sensor data are pre-collected, processed and stored as facts in the recognition system. Rules identify patterns among these facts, which correspond to significant sensor events that may be linked to a specific activity. Instead of searching for patterns arbitrarily, rules exploit the fact that sensors report data periodically; patterns are identified in windows of time that correspond roughly to each sensor's reporting period.

The rule hierarchy of the ADL recognition approach is shown in Fig 2. At the lowest level, rules rely on sensor events to derive atomic activities included in the ADL ontology, as well as location information. An intermediate level involves rules that refine initial derivations and fuse different sources of location information. Then, second and higher level rules progressively combine already recognised activities to infer complex events of increasing complexity. The defined rules rely on information reported from most environmental sensors in the SPHERE house, apart from the temperature, humidity, ambient noise and dust sensors: collected data from these sensors did not yield any AR-related patterns. Furthermore, ambient light sensors proved useful only when the effect of sunlight is minimal, i.e. when the sun is below the horizon.

The rest of this section analyses the rule base of the proposed approach, presenting rules within each distinct category in Fig 2. Rules are expressed in a simplified syntax, where comma denotes conjunction, `=>` denotes inference, `NOT` denotes negation-as-failure, while assert, retract and modify correspond to the typical fact base manipulation actions; in the case of modify, value change is denoted as `valuebefore->valueafter`. Facts are represented as predicates, starting with a capital letter; sensor data are modeled as functions, in capitals, and constant names are in lower-case letters.

3.1 Door Interaction

Door contact (DC) sensors report a zero value while the door is open and a positive value while it is shut. Reports happen either instantaneously or at a period of 10 seconds. Hence, activities that involve a user interacting with a

[4] `http://boxlab.wikispaces.com/Activity+Labels`
[5] `http://oboedit.org`

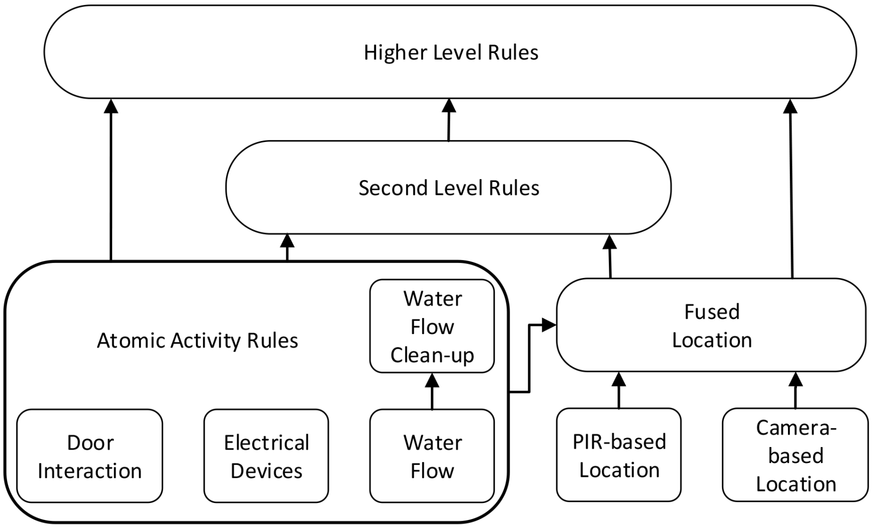

Fig. 2. Hierarchy of ADL recognition rules

door are recognised based on a change in the reported value. The first two rules below detect one or more change events within the sensor reporting window; the third ensures that all activity times refer to the earliest time when the DC sensor reported a zero value. Rules recognising door closing are defined equivalently.

```
DC(t1)=0, DC(t2)>0, (t2-t1)<window => assert(OpenDoor(t2))
DC(t1)=0, DC(t2)>0, (t2-t1)<window, OpenDoor(t3), (t3-t2)<window,
    CloseDoor(t4), t2<t4<t3 => assert(OpenDoor(t2))
DC(t1)=0, DC(t2)>0, (t2-t1)<window, OpenDoor(t3), (t3-t2)<window,
    NOT(CloseDoor(t4), t2<t4<t3) => modify(OpenDoor(t3->t2)
```

3.2 Electrical Devices

Smart meters fitted to electrical devices report consumption every 6 seconds. We can assume, with acceptable accuracy, that a device is switched on when the associated sensor starts reporting positive values. A pair of the rules that follow is defined for every meter-fitted device in the SPHERE house, which includes a TV, microwave, kettle, toaster and fridge. Variants of the first rule are also defined for devices that can be put on standby, such as the TV; turning on from standby is recognised when power consumption increases from a range of positive, non-zero values that correspond to standby consumption. In the case of the fridge, the recognised activities involve opening or closing the fridge door.

```
POWER(device,t1)=0, POWER(device,t2)>0,
    (t2-t1)<window => assert(SwitchOn(device,t2))
POWER(device,t1)>0, POWER(device,t2)=0,
    (t2-t1)<window => assert(SwitchOff(device,t2))
```

3.3 Water Flow

Water meters report the volume of cold or hot water flow instantaneously and while the flow continues but, in contrast to other sensors, they do not report periodically after water flow has stopped. To address this, we follow a two-step approach to recognising water-related atomic activities. The rules below recognise all reports of water flow activity:

```
FLOW(tap,room,t1)>0 => assert(OpenTap(tap,room,t1))
FLOW(tap,room,t1)=0 => assert(CloseTap(tap,room,t1))
```

A pair of these rules is defined for all taps, hot and cold. Then, a second set of rules 'cleans up' the initially recognised events, keeping only the earliest event for each distinct occurrence. The rule for cleaning up open tap events follows; the rule for close tap events is defined accordingly.

```
OpenTap(tap,room,t1), OpenTap(tap,room,t2), t1<t2,
    NOT(CloseTap(tap,room,t3), t1<t3<t2)
    => retract(OpenTap(tap,room,t2))
```

3.4 Complex Activities

Combining the activities recognised by the rules presented so far, we can recognise activities of progressively higher complexity, constructing them recursively. To express the rules, we use a subset of the event algebra defined in [5], with $\wedge$, $\vee$ and NOT denoting conjunction, disjunction and negation-as-failure, respectively and SET denoting unordered sequences of activities, following each other within a maximum time interval. All RHS in the rules imply an assert action.

Atomic activities referring to electrical appliances can be combined to create a complex activity that denotes use of the appliance. The rules below recognise such activities for all devices, with the second inferring a specially named fact for watching TV.

$$SwitchOn(device, t1) \wedge SwitchOff(device, t2) \Rightarrow Use(device, t1, t2)$$
$$SwitchOn(tv, t1) \wedge SwitchOff(tv, t2) \Rightarrow WatchingTV(t1, t2)$$

In the case of activities that involve the use of water taps, the following rules infer possible complex activities. Note that the room associated with each tap influences which activities are recognised.

$$\begin{aligned}
&OpenTap(tap, room, t1) \wedge CloseTap(tap, room, t2) \wedge (t1 < t2) \wedge \\
&\quad NOT(CloseTap(tap, room, t3) \wedge (t1 < t3 < t2)) \\
&\quad \Rightarrow WashHands(t1, t2) \vee WashFace(t1, t2) \\
&OpenTap(tap, bathroom, t1) \wedge CloseTap(tap, bathroom, t2) \wedge (t1 < t2) \wedge \\
&\quad NOT(CloseTap(tap, bathroom, t3) \wedge (t1 < t3 < t2)) \\
&\quad \Rightarrow BrushTeeth(t1, t2) \vee BathingShowering(t1, t2)
\end{aligned}$$

In absence of further information, we cannot discard any of the inferred activities. The second-level complex activities can, in turn, be combined to infer third-level complex activities, such as a user preparing a drink or a snack:

$$SET(Use(kettle, t1, t2), CloseTap(tap, kitchen, t3))$$
$$\Rightarrow PreparingDrink(min(t1, t3), max(t2, t3))$$
$$Use(fridge, t1, t2) \vee Use(toaster, t3, t4)$$
$$\Rightarrow PreparingSnack(min(t1, t3), max(t2, t4))$$

Complex activities can also be inferred using location information, as evidenced from the following rule, which recognises the user walking from one room to another through open doors.

$$IsIn(room, t1, t2) \wedge IsIn(room2, t3, t4) \wedge t2 < t3 \wedge NOT(IsIn(room3, t5, t6) \wedge$$
$$t2 < t5 \wedge t6 < t3 \wedge OpenDoor(t7) \wedge t2 < t7 < t3) \Rightarrow WalkThroughDoors(t2, t3)$$

Note that the IsIn fact refers to the fused location information, as inferred by the rules in Sect. 3.7. Recursive construction of complex events can continue as long as there is a meaningful connection between already recognised events. The next rule recognises the fourth-level complex activity of washing the dishes:

$$(PreparingDrink(t1, t2) \vee PreparingSnack(t3, t4)) \wedge OpenTap(tap, kitchen, t5)$$
$$\wedge CloseTap(tap, kitchen, t6) \wedge min(t1, t3) < t5 < t6 < max(t2, t4)$$
$$\Rightarrow WashDishes(t5, t6)$$

3.5 PIR-based Location

ADL recognition is inextricably linked with the challenge of indoor localisation. In our approach, location information is derived from three sources: PIR sensors, video cameras and recognised atomic activities. This combination is sufficient only for single residential scenarios. The integration of wearable data, which would allow distinguishing between inhabitants is still in progress so, for the remainder of this section, we assume that all inferences refer to the same user.

The PIR sensor reports instantaneously a value of 1, when motion is detected, or 0 otherwise. Based on this, room-level location for single residential settings is inferred as follows:

```
PIR(room,t1)=1, PIR(room,t2)=0, t1<t2,
    NOT(PIR(room,t3)=0, t1<t3<t2) => assert(IsInP(room,t1,t2))
IsInP(room,t1,t2), IsInP(room,t3,t4), t3-t2<=threshold
    => modify(IsInP(t2->t4)), retract(IsInP(room,t3,t4))
IsInP(room,t1,t2), IsInP(room,t3,t4), t3-t2>threshold,
    NOT(IsInP(room2,t5,t6), t2<t5<t3), NOT(IsInP(room2,t7,t8), t2<t8<t3),
    => modify(IsInP(t2->t4)), retract(IsInP(room,t3,t4))
```

The first rule places a user in a specific room, if the corresponding PIR sensor is activated and subsequently deactivated. The next rules merge PIR activation periods in the same room by examining the temporal distance between them. If the distance does not exceed a specified threshold (e.g. roughly 60 seconds), then the periods are immediately considered temporally adjacent and are merged. In the opposite case, we need to ensure that no PIR sensor has been activated in a different room during that gap, before proceeding with the merge.

3.6 Video-based Location

The second source of location information comes in the form of 2D and 3D bounding boxes detected and reported by video cameras installed in the SPHERE house. Each bounding box (BB) is linked to a specific frame id and a user id, to differentiate between boxes in a single frame. It should be stressed that cameras are only installed in the living room, kitchen and main hallway and that rules only rely on 2D and 3D coordinates, which do not carry any sensitive data whatsoever. For single residential settings, the following rules apply:

```
BB(room,frameid,userid,t1), NOT(BB(room,frameid2,userid,t2),
    frameid2= frameid-1) => assert (BBStart(room,userid,t1))
BB(room,frameid,userid,t1), NOT(BB(room,frameid2,userid,t2),
    frameid2= frameid+1) => assert (BBEnd(room,userid,t1))
BBStart(room,userid,t1), BBEnd(room,userid,t2),
    NOT(BBStart(room,userid,t3), t1<t3<t2)
    => assert(IsInV(room,t1,t2), retract(BBStart(), BBEnd())
```

The first two rules detect starting and ending points for bounding box sequences, while the third rule places the user in the room associated with such a sequence. Note that sequences can be merged using the rules defined in Sect. 3.5.

Ghost Sequences It is unavoidable for a video camera tracking body motion to report bounding boxes that do not correspond to an actual user or object, despite efforts in human detection research [9]. Common causes include lingering images that persist after the user has moved or vibrations applied directly or indirectly to the camera. These so-called ghost sequences can severely compromise the validity of video-based indoor localisation, even to the point that fusing other sources is not enough to filter the generated noise. Given that, it makes sense to invest effort in detecting and removing ghost sequences.

After analysing a wealth of available video camera data, we defined a set of ghost detecting heuristics that are applicable to any dataset, especially ones produced using the OpenNI Framework[6]. The simplest heuristic involves discarding any sequence of length below a minimum threshold (e.g. 30 frames, equivalent to 1 second). Integrating this heuristic into the third video-related rule above simply requires adding the conjunct `t2-t1 < threshold`. To deal with the case of ghosts caused by lingering images, the 2D bounding box coordinates are examined. If they remain fixed for longer than a maximum threshold (e.g. again 30 frames), then this stuck subset of the bounding box sequence is discarded. In cases where the user is actually not moving at all, we merge back the two sub-sequences that were separated by removing the stuck subset.

Other ghost detecting heuristics involve examining the 2D bounding box coordinates, along with 3D depth information. If either the width or height of the box is consistently and unjustifiably small, in correlation with depth, then it does not correspond to actual human motion. Finally, application-specific

[6] http://structure.io/openni

heuristics can be considered during ghost detection; for instance, heuristics for the SPHERE house include discarding specific ranges of coordinates that are known to be generated due to surrounding vibrations.

3.7 Fused Location

Having inferred location from PIR and video sensors, the final task is to fuse them into a coherent narrative for room-level indoor localisation. To be able to distinguish between actual and possibly noisy location reports, we associate a confidence value to each PIR sequence (IsInP facts) and each bounding box sequence (IsInV facts). For PIR, confidence is inversely proportional to the number of PIR sensors reporting motion. For video sequences, it depends on the probability of being a ghost, based on the heuristics defined in Sect. 3.6; A sub-sequence is flagged as a ghost while its confidence remains below a specific threshold.

The fusion process essentially infers a single location at any point in time, by combining all available sources using the rules that follow:

- If only a single source reports a location, it is assumed to hold (with a confidence level relative to the associated value)
- If both PIR and video data report the same location, it is assumed to hold (with a confidence value equal to the sum of the individual values)
- If PIR and video disagree, the correct location is the one associated with a recognised atomic activity
- If both disagreeing reports (or neither) are supported by an activity, we assume the report with the higher confidence holds (if equal, we trust PIR).

The result is an ordered temporal sequence of room-level locations, annotated with confidence values that reflect the level of agreement between the various sources. In all cases, rules take into account all possible temporal relations between two sequences, as defined by Allen's interval algebra [1].

4 Real-Time ADL Recognition

The approach presented in the previous section relies on the existence of pre-collected sensor data for the complete period of interest for ADL recognition. While the offline version can assist in diagnosing and managing healthcare and wellbeing conditions, it is unable to provide support for scenarios where emergency assistance is required. In such use cases, activities should be immediately recognised as soon as the associated sensor events take place.

To convert the offline system to a real-time one, a significant change in the nature of both rule and fact bases is required. Instead of representing the history of sensor events, facts now represent the state of each distinct sensor. For each new sensor event, the corresponding fact is modified to reflect the current state. To detect state change, each fact stores the previous state as well. In the rest of this section, we present the required adaptations to the rule base. Note that these are necessary only at the lowest level; all second and higher-level rules remain the same, since they are transparent to the way sensor events are generated.

4.1 Environmental Sensors

The state-based approach for the real-time system simplifies the definition of rules: any state change event is linked to a related atomic activity. This holds for DC sensors, electricity and water flow meters:

```
DCSensor(room,value,prev,t), value=0, prev>0
    => assert(OpenDoor(room,t))
ElecMeter(device,value,prev,t), value>0, prev=0
    => assert(SwitchOn(device,t))
FlowMeter(tap,value,prev,t), value>0, prev=0
    => assert(OpenTap(tap,t))
```

Note that there is no need, as was in the offline case, to clean up duplicate door or tap-related events; these rules fire only once when sensor values change.

4.2 PIR-based Location

In the real-time approach, each consecutive activation/deactivation of a PIR sensor corresponds to the user being in the associated room:

```
PIRSensor(room,value,prev,t), value=1, prev=0
    => assert(PIROn(room,t))
PIRSensor(room,value,prev,t), value=0, prev=1
    => assert(PIROff(room,t))
PIROn(room,t1), PIROff(room,t2) => assert(IsInP(room,t1,t2)),
    retract(PIROn(room,t1)), retract (PIROff(room,t2))
```

Note that since PIROn/Off facts are generated and consumed in real-time, there is no need to check whether they are consecutive: if there was any other such event in between, the IsInP rule would have fired upon assertion. To decide whether subsequent activations extend the user's stay in the room, the following process is carried out (the corresponding rules are not shown for brevity):

- If activation directly follows the last deactivation, we extend immediately.
- If the elapsed time from deactivation to subsequent activation does not exceed a specified threshold, we proceed with the extension (similarly to the second rule in Section 3.5).
- If, in the meanwhile, no activation has taken place in a different room, we extend the already recognised period.
- If the elapsed time is greater than the threshold and there has been an activation in a different room in between, then the new activation is the beginning of a new period of stay in the room.

4.3 Video-based Location

While the other sensors broadcast a single value, video cameras post a wealth of information, which means the state-based approach is not easily applicable; instead, each reported bounding box is stored briefly, only to be combined in facts that represent a period of time during which the user was in the room:

```
BB(room,frameid,userid,t1), NOT(IsInV(room,t2,t3,frame2,frame3),
    frame3=frameid-1) => IsInV(room,t1,t1,frameid,frameid)
BB(room,frameid,userid,t1), IsInV(room,t2,t3,frame2,frame3),
    frame3=frameid-1 => modify(IsInV(room,t3->t1,frame3->frameid))
```

The same ghost detection heuristics, as in the offline mode, are applied; a running confidence value is associated with each sequence, representing the likelihood that it is not a ghost sequence at each point in time.

4.4 Fused Location

In contrast to the offline mode, PIR sequences are not assigned confidence values relative to the number of simultaneous sequences; instead, each time a PIR sensor is activated, the system fuses available video or activity information to decide on its validity:

- If there is no active video sequence and no activity detected, there is no other choice but to assume the user caused the PIR activation.
- If the active video sequence with the highest confidence agrees with PIR, we conclude the user is in the room.
- If video reports a different room, we assume the user is in the room where the most recently recognised atomic activity was performed.

Based on these rules, we can infer room-level location for the user at any given time. Additionally, location history can also be deduced (similarly to the way the offline system reports location), provided that the previous location is stored whenever the user moves to a different room.

5 Experimental Evaluation

5.1 Implementation

Both offline and real-time modes of the ADL recognition system, analysed in Sections 3 and 4, have been implemented in Java, using Jess [7] as a rule engine. Sensor data, which are broadcast and stored in a JSON format, are converted to Java objects, which are then connected to Jess shadow facts. The rule base was divided into several Jess modules, one for each rectangle in Fig. 2. The implemented versions of rules are designed to accommodate variable reporting periods for the sensors in the SPHERE house, since collected data indicated multiple occurrences of early or late reports.

The real-time version is built as an MQTT[7] client, since the SPHERE sensor gateways broadcast data using the MQTT protocol. In order to make sure that no sensor messages are lost, they are processed in separate threads. Whenever a new message is broadcast, it triggers an update in both the Java object associated with the sensor and the corresponding Jess shadow fact.

[7] http://mqtt.org

5.2 Data Collection

To evaluate the ADL recognition system, we used single-occupant, script-based datasets collected in the SPHERE house. Data collection involved 10 participants executing an ADL script of half-hour duration, twice. Participants were asked to visit all house locations which allowed us to observe sensor activations, temporal relationships, and so on. Recognition experiments focused on the following activity categories (a subset of the ADL ontology), included in the script: door interaction, electrical appliance interaction, water tap interaction, preparing a snack or a drink, washing hands/dishes, brushing teeth and bathing/showering. During the experiments, ground-truth data was acquired through annotation of video images collected using a head-mounted, wide-angle, 4K resolution camera. More information on data collection and video annotation can be found in [15].

5.3 Experiments Setup

The evaluation was performed on a Windows® 7 64-bit system powered by an Intel® Core^{TM} i5-2320 processor at 3.00GHz, with 8 GB RAM. For the real-time version, we created an MQTT server and clients, to simulate the SPHERE Home Gateway and sensor gateways, respectively (see Fig. 1). Client simulators parse precollected data and broadcast one sensor message every 5msec, one-third of the camera reporting period, the shortest out of all sensors.

The experiments focus on three aspects: performance, in terms of execution time and memory consumption for the offline mode, activity recognition accuracy, in terms of precision and recall, and localisation accuracy, i.e. the percentage of the experiment duration during which the correct room the user is in is inferred. Precision and recall are commonly defined as $precision = \frac{TP}{TP+FP}\%$ and $recall = \frac{TP}{TP+FN}\%$, where TP, FP, and FN represent activities performed and recognised, recognised but not performed and performed but not recognised, respectively. Precision and recall values are the same for both offline and real-time modes, since only the way of receiving raw sensor data changes.

5.4 Evaluation Results

The results shown in Table 1 are an average of the two times each participant performed the ADL scenario. Also, execution time and memory values are an average of 10 runs for each experiment. Performance results show that the offline version is capable of quickly processing 30 minutes worth of sensor information in 38 seconds, while requiring 170MB, on average. Note that, in real-time mode, recognition delay is negligible due to always maintaining a small fact base.

As far as activity recognition accuracy is concerned, the proposed system shows excellent recall levels of 94.887% on average, while precision is at the somewhat lower level of 87.991%. This is due to the fact that, in cases where available information is not enough to distinguish between a number of possible activities, the defined rules infer them all; this ensures that all performed activities are recognised (higher recall), at the expense of recognising activities that were not performed (lower precision). Finally, the recognition system infers the correct room the user is in 92.715% of the time on average, proving the effectiveness of both ghost detection heuristics and location fusion rules.

Table 1. Results of the experimental evaluation

Experi-ment	**Exec. Time** (Offline) (s)	**Memory** (Offline) (MB)	**Activity Recognition** FP	FN	TP	Precision (%)	Recall (%)	**Localisation** Accuracy (%)
1	32.045	232.4	9.5	4.5	54	85.04	92.31	97.68
2	55.907	200.8	9.5	2	55	85.27	96.49	88.44
3	45.922	129.2	7	1.5	50	87.72	97.09	88.37
4	30.808	157.8	7.5	1	44.5	85.57	97.80	97.69
5	24.548	152.6	7	4	47.5	87.15	92.23	96.14
6	32.642	234.6	9	1	48	84.21	97.96	84.35
7	67.838	179.8	4	4	54	93.10	93.10	90.91
8	26.626	186.4	4	2	51	92.72	96.23	90.90
9	29.615	79.2	3	4	41	93.18	91.11	97.14
10	36.593	149.4	8	3	52	85.95	94.55	95.53
Average	**38.254**	**170.22**	**6.9**	**2.7**	**49.7**	**87.991**	**94.887**	**92.715**

6 Related Work

There has been a substantial amount of research effort on activity recognition, ranging from video-based to sensor-based, and data-driven to knowledge-driven approaches. In the rest of this section, we focus on a selective subset that is more relevant to our approach, presenting the most recent and noteworthy ADL recognition approaches that incorporate logical modelling and reasoning.

Chen et al. [4] model both sensors and activities using ontologies and perform ADL recognition via equivalence and subsumption reasoning on these models. Both offline and real-time modes are supported, while recognition becomes incrementally specific, as more and more sensors are activated. Compared to our approach, this work fails to recognise atomic or lower-level activities unless higher-level ones are recognised. Also, the evaluation scenario is unrealistic, requiring users to perform activities in a predefined, strictly sequential order and fixed time intervals. Finally, the real-time system has a recognition delay of 2-3 seconds, which is significantly slower compared to our approach.

The COSAR system [12] proposes the integration of statistical and ontological reasoning to overcome the limitations of each approach. The statistical component incorporates historical predictions, while the ontological component filters recognitions based on the user's location. Helaoui et al. [6] propose a more tightly-coupled variant, employing a probabilistic DL reasoner. As in our approach, ADL recognition is carried out in multiple levels, building from atomic gestures towards increasingly complex activities; however, apart from the fact that the reasoner requires training data, it is also unable to reason about temporal features and works only in offline mode; also, our activity recognition system consistently outperforms these approaches, in terms of both precision and recall.

Similarly to COSAR, Skarlatidis et al. [13] extend previous work [2] on event calculus-based ADL recognition with probabilistic reasoning based on Markov Logic Networks. Experimental evaluation shows the superiority of the hybrid approach compared to purely probabilistic or event calculus ones, both in terms of recognition rates and robustness against missing data. However, their experiments focus only on posture and movement-related activities as opposed to complex ADL scenarios; also, the intervals of recognised activities are not stored, precluding the ability of inferring activities of higher complexity.

MetaQ [11] is a SPARQL-based reasoning framework for ADL recognition that relies on pattern-based descriptions of both atomic and complex activities. Sensor data are transformed into RDF graphs and native OWL reasoning is performed as an initial classification step. Then, SPARQL queries are produced based on the patterns and are applied on the graphs to realise ADL recognition. In contrast to MetaQ, our approach achieves higher recall while maintaining comparable precision levels; it also includes rules that take into account missing activities and can provide real-time inference of recognised activities.

The work presented in this paper is influenced by previous work [5] that proposed a rule-based ADL recognition system for hierarchically organised and logically consistent complex activities. However, while [5] assumes that atomic activities are already recognised and are given as input to the recognition system, our approach assumes only raw sensor data as input and rules are defined for recognition of both atomic and complex activities. Also, we focus on inferring all possible activities, in both offline and real-time settings.

7 Conclusions and Future Work

In this paper we proposed a rule-based ADL recognition system for multi-modal smart home environments that exploits a bottom-up multi-level reasoning approach to infer events of increasing complexity. The system can operate both on historical and real-time data and exploits the existence of multiple sources to achieve robustness against noise and non-deterministic activity patterns. Experiments conducted in an actual smart home setting used as a testbed prove the effectiveness of the approach and its ability to support AAL scenarios either for long-term monitoring to diagnose and manage health and wellbeing conditions or for directly assisting smart home inhabitants.

Future work involves integrating wearable sensor data to achieve three major objectives: to infer activities unidentifiable with only the other sensors; to improve localisation accuracy or provide an alternate source of location data, in scenarios where privacy is deemed more important than convenience (opting to carry or wear a device, rather than allowing cameras); to explore more complex ADL scenarios with multiple inhabitants and achieve inference of the person performing a recognised activity. Finally, we plan to address scalability issues when faced with increased amounts of sensor input, by exploring methods such as conflict detection and resolution, compression and distributed inference units.

Acknowledgments. This work was performed under the SPHERE IRC, funded by the UK Engineering and Physical Sciences Research Council (EPSRC), Grant EP/K031910/1.

References

1. Allen, J.F.: Maintaining Knowledge about Temporal Intervals. Commun. ACM 26(11), 832–843 (1983)
2. Artikis, A., Sergot, M.J., Paliouras, G.: A Logic Programming Approach to Activity Recognition. In: Scherp, A., Jain, R., Kankanhalli, M.S., Mezaris, V. (eds.) Proceedings of the 2nd ACM International Workshop on Events in Multimedia. pp. 3–8. EiMM '10, ACM, New York, NY, USA (2010)
3. Chen, L., Khalil, I.: Activity recognition: Approaches, Practices and Trends. In: Activity Recognition in Pervasive Intelligent Environments, pp. 1–31. Atlantis Press, Paris (2011)
4. Chen, L., Nugent, C.D., Wang, H.: A Knowledge-Driven Approach to Activity Recognition in Smart Homes. IEEE Trans. Knowl. Data Eng. 24(6), 961–974 (2012)
5. Filippaki, C., Antoniou, G., Tsamardinos, I.: Using Constraint Optimization for Conflict Resolution and Detail Control in Activity Recognition. In: Keyson, D.V., Maher, M.L., Streitz, N., Cheok, A., Augusto, J.C., Wichert, R., Englebienne, G., Aghajan, H., Kröse, B.J.A. (eds.) Ambient Intelligence, LNCS, vol. 7040, pp. 51–60. Springer Berlin Heidelberg (2011)
6. Helaoui, R., Riboni, D., Stuckenschmidt, H.: A Probabilistic Ontological Framework for the Recognition of Multilevel Human Activities. In: Mattern, F., Santini, S., Canny, J.F., Langheinrich, M., Rekimoto, J. (eds.) UbiComp '13. pp. 345–354. ACM (2013)
7. Hill, E.F.: Jess in Action: Java Rule-Based Systems. Manning Publications Co., Greenwich, CT, USA (2003)
8. Kyriazakos, S., Mihaylov, M., Anggorojati, B., Mihovska, A., Craciunescu, R., Fratu, O., Prasad, R.: eWALL: An Intelligent Caring Home Environment Offering Personalized Context-Aware Applications Based on Advanced Sensing. Wireless Pers. Commun. 87(3), 1093–1111 (2016)
9. Liu, J., Zhang, G., Liu, Y., Tian, L., Chen, Y.Q.: An ultra-fast human detection method for color-depth camera. J. Vis. Commun. Image Represent. 31, 177–185 (2015)
10. Maekawa, T., Yanagisawa, Y., Kishino, Y., Ishiguro, K., Kamei, K., Sakurai, Y., Okadome, T.: Object-Based Activity Recognition with Heterogeneous Sensors on Wrist. In: Florén, P., Krüger, A., Spasojevic, M. (eds.) Pervasive Computing. LNCS, vol. 6030, pp. 246–264. Springer (2010)
11. Meditskos, G., Dasiopoulou, S., Kompatsiaris, I.: MetaQ: A knowledge-driven framework for context-aware activity recognition combining SPARQL and OWL 2 activity patterns. Pervasive Mob. Comput. 25, 104–124 (2016)
12. Riboni, D., Bettini, C.: COSAR: hybrid reasoning for context-aware activity recognition. Pers. Ubiquit. Comput. 15(3), 271–289 (2011)
13. Skarlatidis, A., Paliouras, G., Artikis, A., Vouros, G.A.: Probabilistic Event Calculus for Event Recognition. ACM Trans. Comput. Log. 16(2), 11:1–11:37 (2015)
14. Woznowski, P., Fafoutis, X., Song, T., Hannuna, S., Camplani, M., Tao, L., Paiement, A., Mellios, E., Haghighi, M., Zhu, N., et al.: A Multi-modal Sensor Infrastructure for Healthcare in a Residential Environment. In: 2015 IEEE International Conference on Communication Workshop. pp. 271–277. IEEE (2015)
15. Woznowski, P., King, R., Harwin, W., Craddock, I.: A Human Activity Recognition Framework for Healthcare Applications: Ontology, Labelling Strategies, and Best Practice. In: Proceedings of the International Conference on Internet of Things and Big Data (IoTBD). pp. 369–377. INSTICC (2016)